AMENDMENTS

Amendments to the Specification (Corrected):

Please add the following new heading after the heading "Background of the Invention" at line 1 of page 2:

FIELD OF THE INVENTION

Please add the following new heading after the top paragraph at lines 7-12 of page 2:

DESCRIPTION OF RELATED ART

Please add the following five new paragraphs after the last paragraph on page 4 (i.e., after line 17 of page 4):

Software configuration via the known "Makefile" has many limitations. Usage of Makefile is sometimes complicated because the user can see not only the necessary part for configuration but also the contents of the entire file. Most IDE (Integrated Developing Environment) tools, such as Microsoft Visual C++, are no exception. Makefiles and IDE tools are perhaps useful to software developers but not to system integrators.

Middleware has many requirements that should be specified as early as possible. For example, the microprocessor core to which the middleware is mapped, compiler options, real-time operating systems (RTOS), speed (response time), and whether it is software-only or hardware-only or a hybrid solution; all these may be selected by system integrators depending upon their system requirements.

The components of middleware are typically parameterizable and therefore a software configuration must accommodate these parameters in a methodical manner. The components and information relating to their parameters or configuration are not always available on the single machine which the system integrator uses for integration; instead, these components may be distributed on multiple networked machine systems.

Middleware requirements have some similarity to the requirements of modern System-on-Chips (SoCs), which involve integration of multiple functions on a single computer chip. Just like a SoCs, the individual middleware components come from disparate authors and environments, and should ideally be reused again and again.

Presently, in some cases, a programmer uses techniques like pragmas, for example, #if, #ifdef, #else, #elif and #endif, to partition code, whereas in some other cases the programmer creates separate files for every condition. This makes a more complicated configuration process for system integrators, makes the middleware inflexible and difficult to scale, and restricts one to a stand-alone machine. Pragmas are directives to the preprocessor of a C compiler and could be said to be conditional compilation directives.

Please replace the paragraph beginning at line 15 of page 5 and ending at line 2 of page 6 with the following amended paragraph:

~~Software configuration via the known "Makefile" has many limitations. Usage of Makefile is sometimes complicated because the user can see not only the necessary part for configuration but also the contents of the entire file. Most IDE (Integrated Developing Environment) tools, such as Microsoft Visual C++, are no exception. Makefiles and IDE tools are perhaps useful to software developers but not to system integrators.~~ The system based on this invention bridges the gap between the system integrator and completed software.

Please delete the paragraph beginning at line 4 of page 6 that starts with "Middleware has many requirements that should be specified as early..."

Please delete the paragraph beginning at line 11 of page 6 that starts with "The components of middleware are typically parameterizable and..."

Please delete the paragraph beginning at line 18 of page 6 and ending at line 2 of page 7 that starts with "Middleware requirements have some similarity to the requirements..."

Please delete the paragraph beginning at line 4 of page 7 that starts with "Presently, in some cases, a programmer uses techniques like..."

Please amend the paragraph beginning at line 14 of page 9 and ending at line 4 of page 10 with the following two paragraphs:

~~A specific configuration tool included in the embodiment uses script that automatically reads the configuration files in the SCML tag set, shows the system integrator some choices, selects other choices based on the system integrator's choices, shows such other choices to the system integrator to make additional choices, and builds software based on the system integrator's choices.~~

~~XML is used because it is flexible and extensible. A key advantage of using XML in the embodiment is the ability to configure in a distributed environment via the use of common web browsers as interfaces between distributed resources and to interface with the system integrator. Thereby, the interfaces are independent of the particular components.~~

A specific configuration tool included in the embodiment uses script that automatically reads the configuration files in the SCML tag set, shows the system integrator some choices, selects other choices based on the system integrator's choices, shows such other choices to the system integrator to make additional choices, and builds software based on the system integrator's choices.

XML is used because it is flexible and extensible. A key advantage of using XML in the embodiment is the ability to configure in a distributed environment via the use of common web browsers as interfaces between distributed resources and to interface with the system integrator. Thereby, the interfaces are independent of the particular components.

Please replace the heading at line 1 of page 11 with the following amended heading:

BRIEF DESCRIPTION OF THE ~~INVENTION~~ DRAWINGS

Please replace the paragraph at lines 1-3 of page 12 with the following amended paragraph:

Figure 5 shows a distributed hardware and software configuration system as an ~~extention~~ extension of the embodiment of the present invention into a ~~destributed~~ distributed environment;

Please replace the paragraph at lines 7-9 of page 12 with the following amended paragraph:

Figure 7 shows another distributed hardware and software configuration system as an ~~extention~~ extension of the embodiment of the present invention into a ~~destributed~~ distributed environment.